

Tarea Enero – Marzo 2024

Pokémon Pikachu



1 Introducción

Es 1997 y acaba de emerger un nuevo género de videojuego: la mascota virtual. Viendo el éxito que ha tenido la marca Pokémon el último año, Nintendo le ha encargado a usted hacer una adaptación de la marca para entrar en este nuevo género. Debido a que los recursos para desarrollar hardware son limitados, el juego debe ser un tanto más sencillo que el de la competencia, lo cual se espera que también lo haga atractivo a jugadores más jóvenes.

El objetivo principal del juego es crear y mantener una relación con una mascota virtual Pikachu, la cual debe tener al menos 4 estados emocionales, los cuales se deben representar con ASCII Art. A continuación se muestran unos estados de ejemplo. **No** se espera que los estados finales de su programa coincidan exactamente con ellos.

Rango de ejemplo	0 – 2500	2500 – 5000	5000 – 7500	7500 – 10000
Relación ejemplo	No quiere ni verte	A regañadientes	Te acepta	Te quiere
Representación de ejemplo	<pre> /\ /\ \ \ / / / \ () \ / _ _ _ </pre>	<pre> /\ /\ \ \ / / / \ \ / () - () \ M / </pre>	<pre> /\ /\ \ \ / / / \ - - () - () \ / _ _ _ </pre>	<pre> /\ /\ \ \ / / / \ ^ ^ () - () \ O / </pre>

La relación siempre empieza en 0 y crece hasta un máximo de diez mil (10000)

2 Requerimientos del programa

Debe crear un programa en ANSI C que simule este dispositivo. El programa debe presentar un menú con las siguientes opciones, acompañado del ASCII art correspondiente al estado emocional actual de Pikachu; y de un prompt para que el usuario escriba el nombre de la opción.

- **Tiempo**, opción que muestra el tiempo total transcurrido desde que se comenzó el juego (medido comparando la hora actual con la hora de inicio usando la función `time` de `time.h`)
- **Watts**, opción que debe mostrar el balance actual del jugador en watts (dinero del juego) el cual debe ser un número entero
- **Tienda**, opción muestra los regalos que se pueden comprar para Pikachu y permite comprarlos
- **Pikachu**, opción que debe mostrar el valor actual de la relación con Pikachu, el nombre del emocional, y la lista de regalos que Pikachu ha recibido
- **Jugar**, opción que debe permitir jugar el siguiente juego con Pikachu:
 - El jugador decide cuántos watts apostar con Pikachu (mientras no exceda su saldo)
 - Pikachu escoge aleatoriamente uno de los regalos que el jugador le ha dado (se le indica al jugador que se ha hecho mediante un mensaje de texto, o mediante un ASCII art, acompañado de un prompt – **no** se le debe decir cuál se escoge)
 - El jugador debe adivinar el regalo, introduciendo el nombre del regalo
 - Si el jugador acierta, se le indica al jugador que Pikachu ha “cargado” su saldo (Pikachu reconoce que perdió), y el jugador recupera los watts apostados duplicados (es decir, si el jugador tenía un balance de n watts, y decide apostar m de esos watts, luego de ganar el juego debe tener un balance de $n+m$ watts)
 - Si el jugador no acierta, se le indica al jugador que Pikachu ha consumido sus watts, los cuales son deducidos de su balance y sumados a los puntos de relación a razón de 1 watt = 1 punto de relación (Pikachu se alegra de haberte ganado)
 - Si el jugador no le ha comprado al menos 2 regalos, Pikachu debe rehusarse a jugar

El programa le da al jugador, por cada segundo transcurrido desde que comenzó a ejecutar el programa, 1 watt. Adicionalmente, le da 10 watts por “usar” el programa (es decir, cada vez que entra en una opción del menú, a excepción de la opción “jugar”).

La relación con Pikachu se degrada a razón de 1 watt por segundo, pero no puede ser menor que cero. Esto es para motivar al jugador a interactuar con Pikachu regularmente e incentivarlo a usar la tienda.

2.1 Tienda

La tienda debe tener los siguientes regalos para darle a Pikachu

Regalo	Costo	Efecto
Baya	100 Watts	+100 Relación
Bayamarga	200 Watts	+250 Relación
Pokeball	300 Watts	+400 Relación
Antiparabaya	400 Watts	+550 Relación
Baya misterio	500 Watts	+700 Relación
Baya milagro	600 Watts	+850 Relación
Baya dorada	700 Watts	+1000 Relación
Baya importada	800 Watts	+1150 Relación
Caramelo Raro	900 Watts	+1300 Relación

Al seleccionar un regalo (introducir su nombre en el prompt seguido de “↵ enter”), se debe

- Deducir su costo del saldo del jugador
- Aumentar la relación en el valor dado
- Agregar el elemento a la lista de regalos que Pikachu ha recibido
- Imprimir un mensaje de confirmación
- Volver a mostrar la lista de regalos disponibles, permitiéndole al jugador comprar otro

2.2 Lista de regalos recibidos

La lista de regalos debe implementarse como una lista enlazada, ya sea simple o doble. Los regalos del mismo tipo deben ser indistinguibles, para propósitos de jugar con Pikachu (así, si la lista contiene [baya, pokeball, baya] y el regalo elegido al azar es el primero, debe comportarse igual a si se eligiera el tercero; en otras palabras, ambos regalos pueden ser adivinados escribiendo “baya”)

2.3 Random

Debe usar la función `rand` de `stdlib.h` para decidir el regalo que escoge Pikachu en el juego. Para ello, debe primero fijar `srand` usando como semilla la hora de inicio del programa. La llamada a `srand` debe ocurrir sólo una vez.

2.4 Entrada y Salida

Todas las entradas deben realizarse por un prompt en el cual el usuario debe escribir el nombre completo de la opción que le interesa, y luego presionar la tecla “↵ enter”. No debe ser sensible a mayúsculas. Si la opción introducida no está en el menú, debe imprimirse un mensaje de error y permitir al usuario introducir otra opción sin volver a imprimir el menú. Todos los menús deben aceptar el comando “volver” para volver al menú anterior (si vuelve al menú principal, se debe actualizar el ASCII art acorde con los últimos cambios en la relación). Si se introduce este comando en el menú principal, el programa se debe cerrar. El programa **no** debe manejar archivos. En particular, el programa **no** debe guardar el estado al salir más allá de lo que se pueda imprimir por la consola.

3 Requerimientos de la entrega

Debe entregar un archivo comprimido que contenga

- Todos sus archivos `.c` con sus respectivos `.h`
- Un Makefile para compilar su programa

El programa debe compilar sin errores al llamar al comando `make`. No incluya archivos `.o` ni archivos ejecutables en su entrega.

Debe documentar su código, incluyendo una breve definición al inicio de cada archivo, función y macro que justifique qué es e indique para qué es cada parámetro; y para cualquier variable cuyo nombre no sea intuitivo.

El proyecto debe ser subido al Moodle de la materia en la sección marcada como “📁 Tarea” en la sección de “Semana 1”. Sólo deberá efectuar una entrega por grupo.

4 Evaluación

El proyecto tiene una ponderación de 10 puntos. Se asignarán

- 5 puntos por código
 - 1 punto por su lista enlazada
 - 1 punto por las funciones que utilizan la lista enlazada
 - 1 punto por su uso correcto de `time.h` y `stdlib.h`
 - 1 punto por su forma de actualizar los watts y la relación
 - 1 punto por su forma de procesar la entrada
- 5 puntos por ejecución
 - 1 punto por la funcionalidad de la tienda y el menú principal
 - 1 punto por mantener el saldo y la relación con Pikachu actualizados correctamente
 - 1 punto por imprimir correctamente la opción Pikachu (incluyendo la lista completa de regalos dados)
 - 1 punto por la jugabilidad del juego “jugar con Pikachu”
 - 1 punto por la originalidad de sus estados emocionales y ASCII art

El programa debe correr sin errores y seguir todas las prácticas indicadas en el laboratorio para la programación de C en Unix.

4.1 Puntos Extra

Se considerará para puntos extra cualquier funcionalidad adicional relacionada con la mascota virtual Pokémon Pikachu o sus sucesores (e.g. Pokéwalker, Poké Ball Plus, etc) que sea incorporada al programa, producto de investigar en internet. Debe mostrarse explícitamente en la ejecución (por ejemplo, imprimiendo la opción en el menú) y estar debidamente documentado en el código, para ser considerado.

En ningún caso se considerará para puntos extra más puntos de los que vale la tarea (es decir, no se podrá más que duplicar los puntos de la tarea)